Application of Spoken and Natural Language Technologies to Lotus Notes Based Messaging and Communication

Stephanie Lajoie, Oliver Carr,
Michael Coleman and Ahmad
Hashemi-Sakhtsari

DSTO-TR-1265

Application of Spoken and Natural Language Technologies to Lotus Notes Based Messaging and Communication

Stephane Lajoie, Oliver Carr, Michael Coleman and Ahmad Hashemi-Sakhtsari

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-TR-1265

ABSTRACT

Speech can potentially provide a natural, efficient and flexible means of user interaction with computers in a messaging and communication environment. This report presents a spoken language data entry and database query concept demonstrator. Firstly, the report discusses a technique for integrating a commercial large-vocabulary continuous speech recogniser with a Lotus Notes database user interface for data entry into Lotus Notes forms. Secondly, the report describes the integration of a number of software components, which include a commercial natural language processing product, in order to produce a query interface to access a Lotus Notes database. The military operators can query a database in plain English to obtain information and to monitor their internal workflow.

RELEASE LIMITATION

Approved for public release

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Edinburgh South Australia 5111 Australia

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR-012-130
January 2002

APPROVED FOR PUBLIC RELEASE

Application of Spoken and Natural Language Technologies to Lotus Notes Based Messaging and Communication

Executive Summary

The application of language technology to produce an interface between humans and computers has a number of potential advantages.

In a survey on military applications of human-machine communication by voice, the most pervasive near-term application was identified as being voice data entry, which can range from entering numerical data, to creating formatted military messages, to free-form report entry. The first aim of this research and development project was to allow operators to use direct speech input to dictate contents of fields in any form from a Lotus Notes database used within the Australian Deployable Joint Force Headquarters (DJFHQ). This was because speech recognition technology can potentially allow users to enter data in a faster and more accurate way than using manual input modality.

Although there have been recent advances in natural language processing, only systems in limited domains can be envisioned in the near term. A good example of this is a simple question-answering domain, such as a database interface, where the performance of natural language processing systems is regarded as being acceptable.

The second aim of the project was to improve users' ability to query Lotus Notes databases. As part of the system requirements, improving the resolution of the data to be searched, reducing the system response time, and increasing the search and data extraction capabilities from a Lotus Notes database were also to be investigated. This was to enable military operators to monitor their internal workflow more efficiently by asking a broad range of questions from the Lotus Notes databases.

A large number of commercial off-the-shelf speech recognition products are now available. These include Dragon NaturallySpeaking, which is a large-vocabulary, speaker-dependent continuous speech recogniser by Dragon Systems™. English Wizard is a natural language processing software product, produced by Linguistic Technology Corporation, which provides the capability of natural text query from a relational database by using natural language processing techniques.

This report presents details of how a concept demonstrator has been developed that allows users of Lotus Notes databases to enter data and make queries from these databases by integrating the two language technologies, Dragon NaturallySpeaking and English Wizard. These databases are used for messaging and communication at

the DJFHQ. Since both Dragon NaturallySpeaking and English Wizard are Commercial-Off-The-Shelf (COTS) products, this report also highlights some of the practical limitations in integrating COTS language technology products with military messaging systems.

DJFHQ is organised into several cells. The Lotus Notes database used for this research is from the Joint Logistics, Personnel and Manpower (J1/4) cell of the headquarters. This cell uses a Lotus Notes database to log and distribute information regarding issues such as transportation, medical and legal matters, and staffing levels. A user will enter the information into the database by entering data into various fields. These fields contain various elements, for example formatted data tokens, attachments and formatted or unformatted text. A database, known as Command and Support System (CSS) database, has been implemented in Lotus Notes by the Australian Defence Industry contractors situated at the DJFHQ. The CSS database has five document types that a user creates, edits and sends. Documents derived from these types can be sent to other cells within the DJFHQ and to other external organisations.

Dragon NaturallySpeaking was integrated with the CSS database. Dragon Systems' scripting commands were used since they provide the quickest and easiest method of integration. Scripting commands control Lotus Notes by simulating mouse, keyboard and Dynamic Data Exchange (DDE) events that are generated when the recogniser hears a word or a phrase that is predefined and has a simple syntax. The scripting language has functions that use Dynamic Data Exchange (DDE) calls to start and close applications and switch between windows in the Windows environment.

The use of the scripting language for speech integration does not require the source code of the application, but one major drawback with scripting commands is context insensitivity in data entry in Lotus Notes forms or fields.

Lotus Notes does not allow for an Open DataBase Connection (ODBC) to a database. In order to achieve a Structured Query Language (SQL) query from a Lotus Notes database, its data had to be moved into a relational database. A Lotus Notes Querying Program was developed as a query interface to a relational form of the Lotus Notes database to provide the users with better searching and data extraction capabilities than what was available. This allowed operators to obtain information and monitor the workflow within their messaging environment.

English Wizard provided functional natural language processing capability by allowing the user to enter a query, as an English expression, either by voice or keyboard. The question in text form is passed into English Wizard and is translated into SQL using definitions from an English Wizard dictionary. The derived English Wizard SQL is executed; the data is recovered from a relational database and is placed into a data grid for viewing.

Custom software is incorporated with English Wizard to provide a user interface to enter a query and display either tabular results or an aggregate result. A series of questions can be loaded and executed, and a question can be saved. The product known as Querying Program allows query results to be displayed within Lotus Notes as a view. A Lotus Notes database view contains a list of documents that meet some criteria.

In order to produce an effective user interface with a speech recognition capability, studying human to human conversations in the application domain is required. An important aspect of this study has been to provide information on how the Australian Defence Force (ADF) staff pronounce specialised words or utter acronyms and jargon terms to others and to the speech user interface. Knowledge about the use of acronyms is crucial in developing vocabularies to deliver a speech recognition capability to military messaging. Information on how acronyms are pronounced has been elicited by observing military operators of Lotus Notes databases using speech for data entry.

Preliminary observations of staff at DJFHQ indicated that direct speech input provides faster, more accurate and natural means of user interaction with the databases, especially for those with limited typing skills.

Many factors affect the performance of a speech recogniser. Amongst these are rate and manner of speaking and the level of articulation. These factors influence a speech recogniser to behave unpredictably, resulting in user frustration. Despite the difficulties, it has become evident that the technology can potentially provide a faster and more accurate means of user interaction with military messaging and communication systems.

It is well known that correcting words or phrases by speech takes longer than dictating them. Although it would often be most efficient to enter data by a combination of speech and manual input modalities, errors should be corrected by either manual input alone or through an interactive correction mechanism by switching between modalities.

The query interface to a Lotus Notes database has enabled users to ask questions in plain English in order to obtain responses of a statistical nature or view documents containing particular information. Although the possible range of allowable queries thus far is limited, this has enabled the operators to monitor their internal workflow in a flexible and efficient manner by allowing them to ask a broad range of questions. It is noted that performing quantitative evaluation of user interaction with a spoken language query or dialogue system is hard due to inherent variabilities present in asking English questions.

One of the limitations of the concept demonstrator is that when the Lotus Notes Query Program creates a view in Lotus Notes as a result of a user query, the view is not updated in the Lotus Notes Client. The user needs to close and reopen the Lotus Notes Client to re-display the view created as a result of a query. Additionally, limited scalability of the commercial software technology used and the security implications of using a text file need to be addressed when further developing the system.

In order to overcome the above-mentioned limitations, it is intended to use Java, namely its Java DataBase Connection (JDBC), to communicate directly with a Lotus Notes database. The communication involves using the Lotus Notes Java classes.

One approach to achieve a Structured Query Language query from a Lotus Notes database could be to use the Lotus Notes C++ Application Programming Interface (API) to move the data into a relational database. Data communication with Lotus Notes can be achieved by using the Lotus Notes C++ API; such an API is also available for Java. Future research and development of the prototype model described in this report will aim to use the Java programming language to improve the user interface and integrate language and database technologies with Lotus Notes databases. It is intended to produce a user interface for a Lotus Notes database that allows the user to create, edit and view entries in the database. The user will be able to enter information by speech as well as keyboard and mouse. The speech integration will be context sensitive and robust. Better searching and querying capabilities from Lotus Notes databases will be provided to the user. Internet browsers (e.g. Internet Explorer) are being used at the DJFHQ to read Lotus Notes databases. Since this graphical user interface will be written in Java it can be used within an Internet browser, such as the Internet Explorer, to allow Lotus Notes databases to be read.

Authors

Stephane Lajoie
Information Technology Division

Stephane Lajoie completed his HSC in Eltham College Victoria between 1984 and 1989. After a period of leave from study, he completed a Diploma in Software Development in the Northern Metropolitan Institute of TAFE, Victoria, during 1996-1997 and he completed his Degree in Computer Science at the Swinburne University, Hawthorn Victoria in the year 2000. He was on industrial placement with Information Technology Division in DSTO during the calendar year 1999.

Oliver G Carr
Information Technology Division

Oliver Carr graduated with a Bachelor of Science in the Faculty of Mathematical and Computer Sciences (B.Sc (Ma & Comp. Sci)) from the University of Adelaide in 1997. Oliver is undertaking a Pure Mathematics Honours Degree part-time as well as working in DSTO in the Information Technology Division as a Professional Officer. His interests include cryptography, integrating Commercial off-the-shelf speech recognisers into military applications, and evaluating large-vocabulary continuous speech recognisers for dictation purposes.

Michael Coleman
Information Technology Division

Mike Coleman holds a Bachelor Degree in Applied Mathematics. He worked as a civilian programmer for the army during 1986. Mike joined DSTO in 1987. While at DSTO he has been involved on several software projects. He worked with Ahmad Hashemi-Sakhtsari, and Oliver Carr on various aspects on the use and integration of commercial speech applications in Defence for approximately one and half year during 1999-2000. Mike is currently pursuing a part-time Postgraduate Diploma in Computer Science at the University of Adelaide.

Ahmad Hashemi-Sakhtsari
Information Technology Division

Ahmad Hashemi-Sakhtsari is a research scientist in Human Systems Integration (HSI) group. His current research work is focused on application of commercial language technology to military systems and on studying human computer interaction through speech as well as manual modalities. He leads a small speech technology research and development team in the HSI group.

Contents

1. Aims

There were two general aims in undertaking this research and development project to integrate spoken language technology with Lotus Notes databases. The first aim was to allow the user to dictate the contents of a field into a Lotus Notes database form, or to select a field for data entry via speech. This would supplement the existing keyboard and mouse selection modalities with a view to providing users with the means to enter data in a faster and more accurate way than through manual input modality alone (Martin and Welch, 1980; Lea, 1983).

The second aim was to improve the users' ability to query Lotus Notes databases. Natural language processing allows users to obtain information from a Lotus Notes database by using queries constructed in plain English. As part of the requirements, the possibility of increasing the search capabilities within Lotus Notes was to be investigated so that operators at the Deployable Joint Force Headquarters can monitor their internal workflow more efficiently than they were able to do using Lotus Notes querying facility alone.

2. Introduction

Lotus Notes is used widely for messaging and communication throughout various headquarters in the Australian Defence Force. The Australian Defence Deployable Joint Force Headquarters (DJFHQ) has requested that language and database technologies be integrated with their Lotus Notes databases and related human computer interaction issues be investigated.

A number of early assessments of military applications of speech recognition technology (Woaded and Cupules, 1983; Flanagan et al., 1984; Makhoul et al., 1989; Cupples and Beek, 1990) provide reviews of the state of the art at that time. Each review outlines a number of programs in which prototype speech recognition systems were tested in application environments, but data entry is the most pervasive application of the technology.

A large number of commercial speech recognition products are now available. These include large-vocabulary speaker-dependent continuous speech recognisers by Dragon Systems™, IBM™, and Lernout and Hauspie Speech Products™. Dragon NaturallySpeaking is one of the leading products in the market. English Wizard is a natural language processing software, produced by Linguistic Technology Corporation, which provides the capability of natural text query from a relational database by using natural language processing techniques.

This report presents a concept demonstrator spoken language data entry and dialogue system by integrating the two language technologies, Dragon NaturallySpeaking and English Wizard, with Lotus Notes databases at the DJFHQ.

As part of the requirements by the DJFHQ, the concept demonstrator allows data entry into Lotus Notes forms by voice and permits the operator to obtain information from a Lotus Notes database by using queries constructed in plain English. Since both Dragon NaturallySpeaking and English Wizard are Commercial-Off-The-Shelf (COTS) products, this report also highlights some of the practical limitations in integrating COTS language products with military messaging systems.

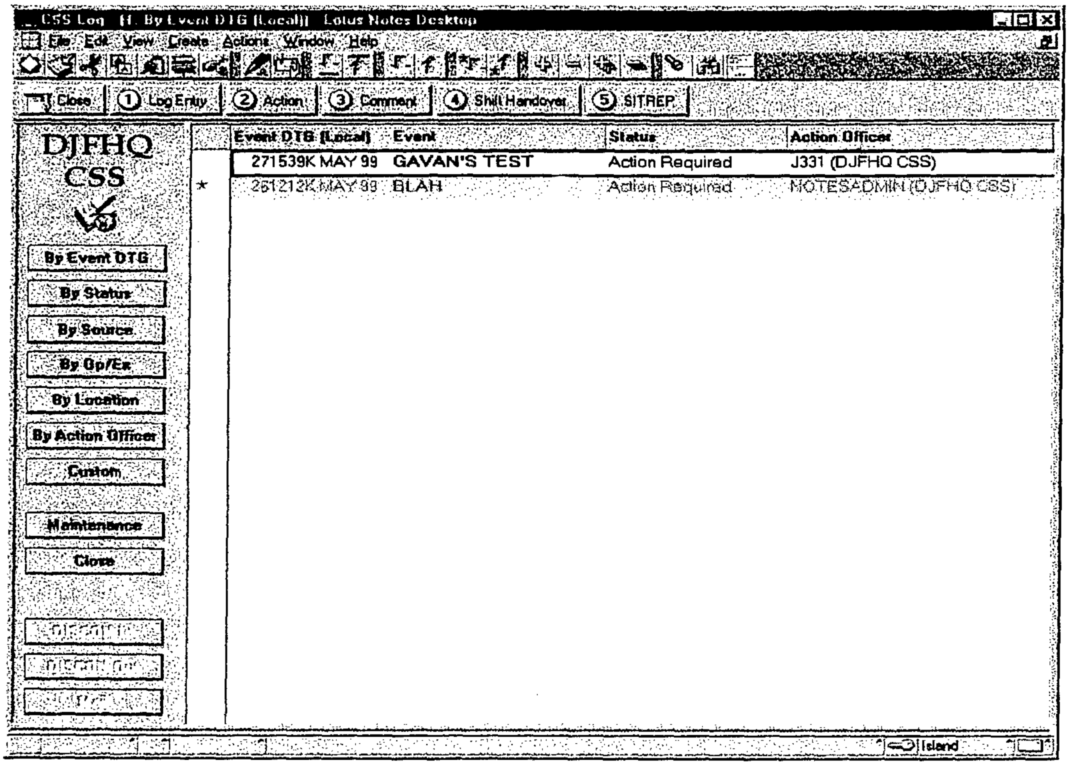
2.1 Use of Lotus Notes in Deployable Joint Force Headquarters

DJFHQ is organised into several cells. The Lotus Notes database in question here is from Joint Logistics, Personnel and Manpower (J1/4) cell of the headquarters. This cell uses a Lotus Notes database to log and distribute information regarding issues such as transportation, medical and legal matters, and staffing levels. A user will enter information into the database by entering data into various fields. These fields contain various elements, for example formatted data tokens, attachments and formatted or unformatted text.

2.2 An Overview of Lotus Notes Command Support System Database

The Australian Defence Industry contractors situated at the DJFHQ have implemented a database known as Command and Support System (CSS) database in Lotus Notes. The CSS database, as shown in figure 1, has five document types that the user creates, edits and sends. The types of documents are Log Entry, Action, Comment, Shift Handover and SITuation REPort (SITREP). Documents derived from these types can be sent to other cells within the DJFHQ and other external organisations.

The Log Entry form is used to log events or activities involving or applicable to the DJFHQ's operational requirements. These logs can range from health situation reports to operational orders. In the Log Entry form there is a field that selects officers to be informed or orders an officer to perform a task. The officer ordered receives a copy of the Log Entry form and generates an Action form describing the outcome and status of the task. The Comment form is used to record additional comments on the outcome and status described in an Action form. The Shift Handover form is used to roster staff and to record the current situation at the time of the shift handover. The left-hand column in figure 1 shows the various views of the database by particular fields. For example, the view of the CSS log database, shown below, is sorted by Event Date Time Group (DTG).



	Event DTG (Local)	Event	Status	Action Officer
	271539K MAY 99	GAVAN'S TEST	Action Required	J331 (DJFHQ CSS)
*	261213K MAY 99	BLAH	Action Required	NOTESADMIN (DJFHQ CSS)

Figure 1: Main View of Command Support System Database

3. Integration of Dragon NaturallySpeaking Speech Recogniser with Lotus Notes Databases for Data Entry

There are a number of advantages in applying language technology to produce an interface between humans and computers (Zue, 1994). Speech is natural; humans speak before they can read and write. Speech is efficient; generally 3 words or 10 phonemes are uttered per second, whereas typing speed rarely exceeds 60 words per minute. Finally speech is flexible; it provides hands-free interaction.

As in human factors engineering or ergonomics, the goal of integrating speech recognition with applications is to increase reliability, performance and user satisfaction while at the same time decreasing stress and discomfort (Kloosterman, 1994).

In a survey on military applications of human-machine communication by voice, the most pervasive near-term application was identified as being voice data entry, which can range from entering numerical data to creating formatted military messages, to free-form report entry (Weinstein 1994).

In order to achieve direct speech input, Dragon NaturallySpeaking from Dragon Systems was integrated with a Lotus Notes database used by the J1/4 cells at the DJFHQ. The quickest and easiest method of integrating Dragon NaturallySpeaking with an application is by using Dragon Systems' scripting commands. The scripting language was developed by Dragon Systems for their earlier speech recogniser, DragonDictate. The scripting commands have not changed and the product is called DragonDictate Macro Language (see DragonDictate Macro Language - Guide and Reference). A scripting command controls a Windows application, such as Lotus Notes, by simulating keyboard, mouse and Dynamic Data Exchange (DDE) events that are sent to the application. These events are generated when the recogniser hears a word or a phrase that is predefined by a simple syntax. In order to use the scripting commands, knowledge of how to navigate an application via keyboard and mouse events is required. Figure 2 below shows how scripting commands simulate what the user can do with the keyboard and mouse. A collection of scripting commands are collated into a file called a script, which can be made application and user specific. A default script can be given to new users of Dragon NaturallySpeaking.

Since the scripting language can only send keyboard, mouse and DDE events and is context insensitive, it can only differentiate between windows that have a unique title. Although a window can be brought into focus, fields from a form in that window cannot be. For example, a word can be mistakenly uttered into a number field, but since the speech recogniser is unable to distinguish which field is in focus, the word will be displayed in the number field. Knowledge about context is required in error recovery since re-training erroneous utterances can automatically be initiated taking context into account.

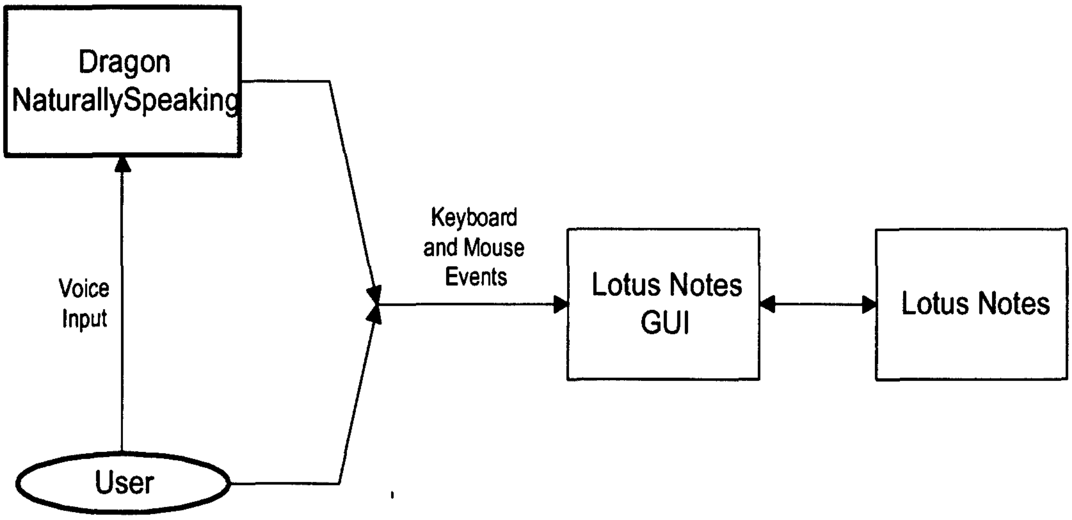
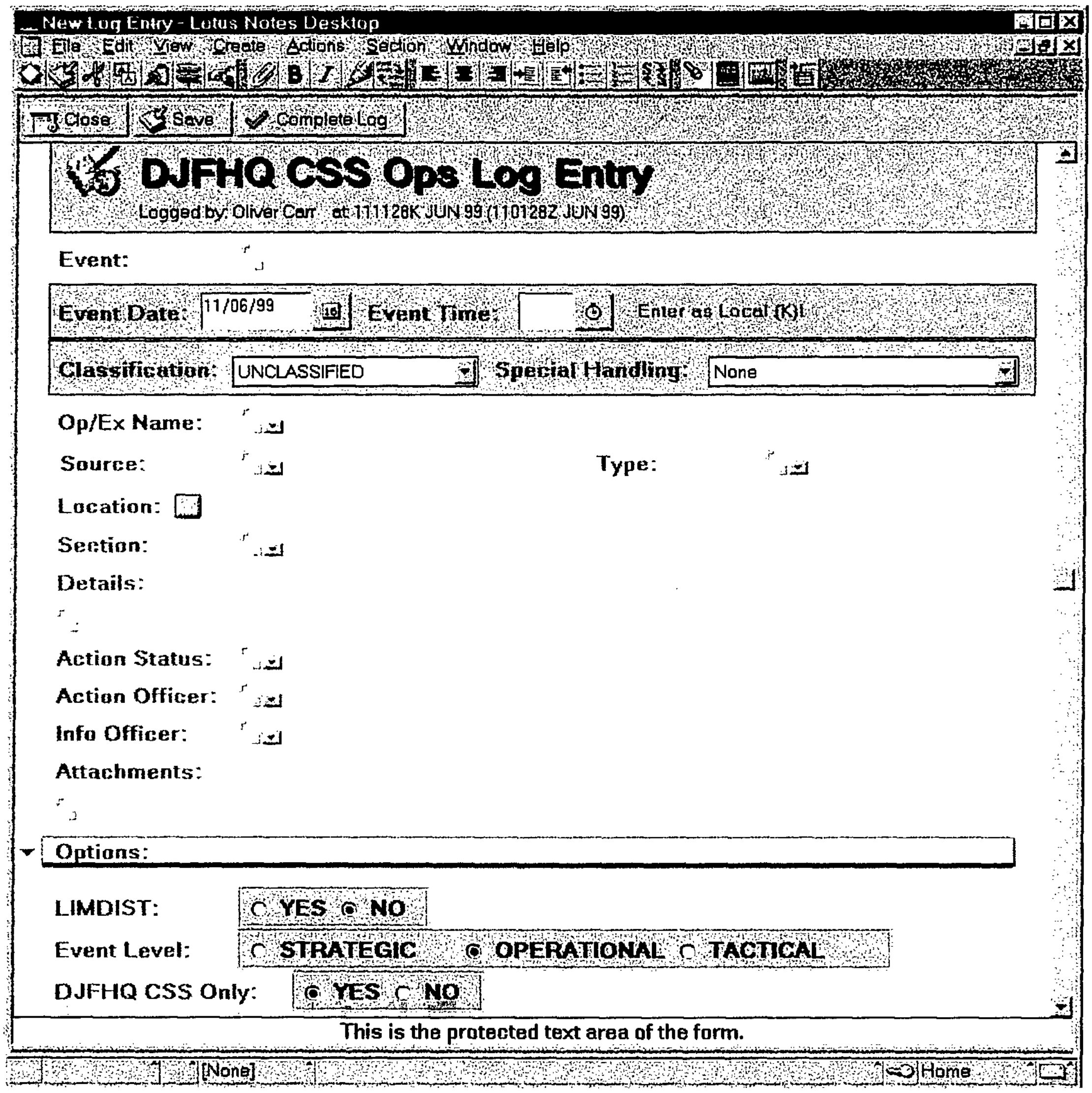


Figure 2: A Block Diagram showing how Dragon Scripting Commands are sent to a Lotus Notes Database to Simulate Keyboard and Mouse Events

The scripting language has functions that use Dynamic Data Exchange (DDE) calls to start and close applications and switch between windows in the Windows environment. The use of the scripting language does not require the source code of an application in order to integrate speech with that application.

3.1 Speech Integration with Command Support System Database through Scripting Commands

As a prototype model, speech recognition technology was integrated with all the fields in each of the five forms in the CSS database using scripting language.


New Log Entry - Lotus Notes Desktop
File Edit View Create Actions Section Window Help
Close
Save
Complete Log

DJFHQ CSS Ops Log Entry

Logged by: Oliver Carr at 111128K JUN 99 (110128Z JUN 99)

Event:

Event Date: 11/06/99 Event Time: Enter as Local (K)!

Classification: UNCLASSIFIED Special Handling: None

Op/Ex Name:

Source: Type:

Location:

Section:

Details:

Action Status:

Action Officer:

Info Officer:

Attachments:

Options:

LIMDIST: ○ YES ● NO

Event Level: ○ STRATEGIC ● OPERATIONAL ○ TACTICAL

DJFHQ CSS Only: ● YES ○ NO

This is the protected text area of the form.

Figure 3: Log Entry Form of Lotus Notes Command Support System Database

Dragon NaturallySpeaking was integrated with all the fields in each of the five forms in the CSS database using the scripting language. The most complex form of the CSS database is the Log Entry form, shown here in figure 3. There are 18 fields in total in the Log Entry form. The second field is the Event Date, which is the date the log entry refers to. As an example of data entry by voice, the user can say "event date 8 11 99" in order to enter a date into this field. The corresponding syntax for this command would be "event date <day><month><year>". The angle brackets containing text signify a parameter. The parameters are in this case all numbers. Upon successful recognition, the command for data entry will initiate the following sequence of keystrokes:

```
{Alt+a}+t
{Tab}
{Shift+End}+{Del}
8/11/99
```

The notation {Shift+End} means the Shift key is held down whilst the End key is pressed. The "+{Del}" after {Shift+End} means the Del key is then pressed. This particular sequence of keystrokes deletes the contents of the Event Date field. Similar scripting commands have been implemented for all the fields in all the forms from the CSS database. For example, if the user presses the Enter key in the 'Source' field, the 'Select Keywords' window opens with the values for the 'Source' field. The recognition accuracy could be improved if the title of this window was specific to the field type and changed for each field in a form in use. To navigate through the Log Entry form the CSS database had to be modified by adding several LotusScript macros. These macros were invoked by a scripting command, such as {Alt+a}+t.

3.2 Speech Integration with Command Support System Database using ActiveX Components

Since the scripting language can only mimic keyboard and mouse events and is context insensitive, it can only differentiate between windows that have unique titles. Although a window can be brought into focus, fields from a form in that window cannot be. Context awareness is required to improve error correction.

Figure 4 shows data flow between an application and a speech recogniser through ActiveX components. These components allow full integration of Dragon NaturallySpeaking with an application. The available functions provided by the ActiveX components include:

- Switching between users and creating new users
- Calling microphone setup program and initiating general training
- Executing Scripting Commands

- Generating custom events for each utterance depending on the current context of the application
- Creating commands for control and navigation within an application
- Dictation into free text fields
- Text to speech synthesis

ActiveX control increases recognition accuracy and improves error correction and recovery. For example, through a two-way communication, the recogniser can inform the application that a date field has been selected by a user. The application can then return a command to the recogniser directing it to only accept valid date entries from an active sub-vocabulary.

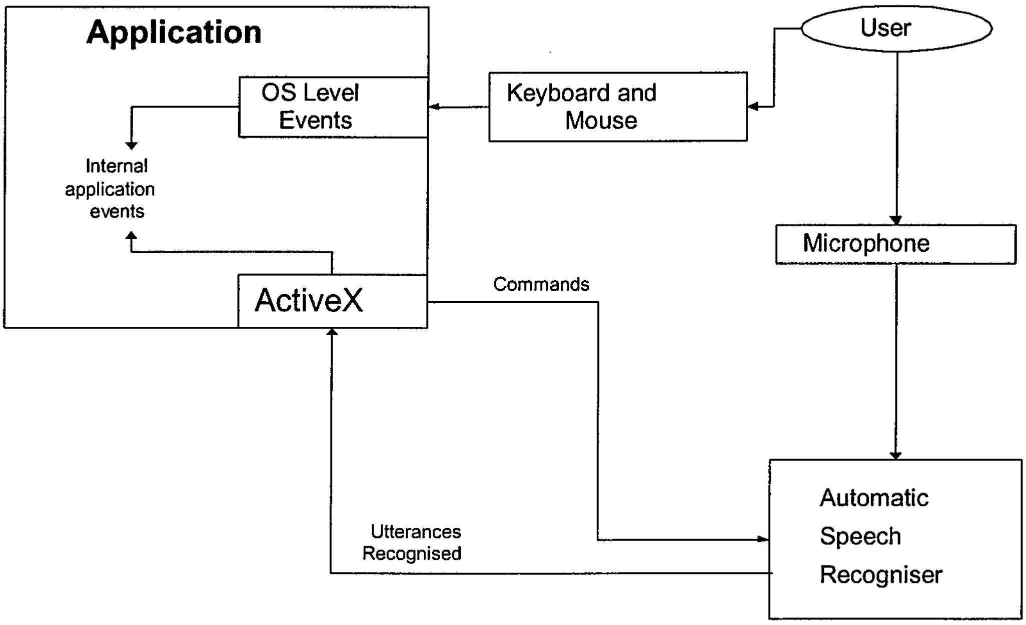


Figure 4: Data Flow between a Speech Recogniser and Closer an Application through ActiveX Components. This Provides Interaction between Application and Speech Recogniser.

The ease of error correction and recovery determines the user acceptance of a speech recogniser. In a recent study (Karat et al., 1999) the most common command used by users of continuous speech recognisers was the command that reversed the immediately preceding action. This command is "Scratch That" with Dragon NaturallySpeaking. Although the users in the study could dictate on average at 107 words per minute, the average correction time took three times as long. It would often be most efficient to enter data by a combination speech and manual input modalities. However, error correction is best done through either manual input alone or through an interactive correction mechanism by switching between the modalities (Suhm et al., 1999).

It is intended to use the ActiveX components available with the Dragon NaturallySpeaking software in the future development for data entry in Lotus Notes databases. This will improve error correction and recovery since the context in which an error is made will be known.

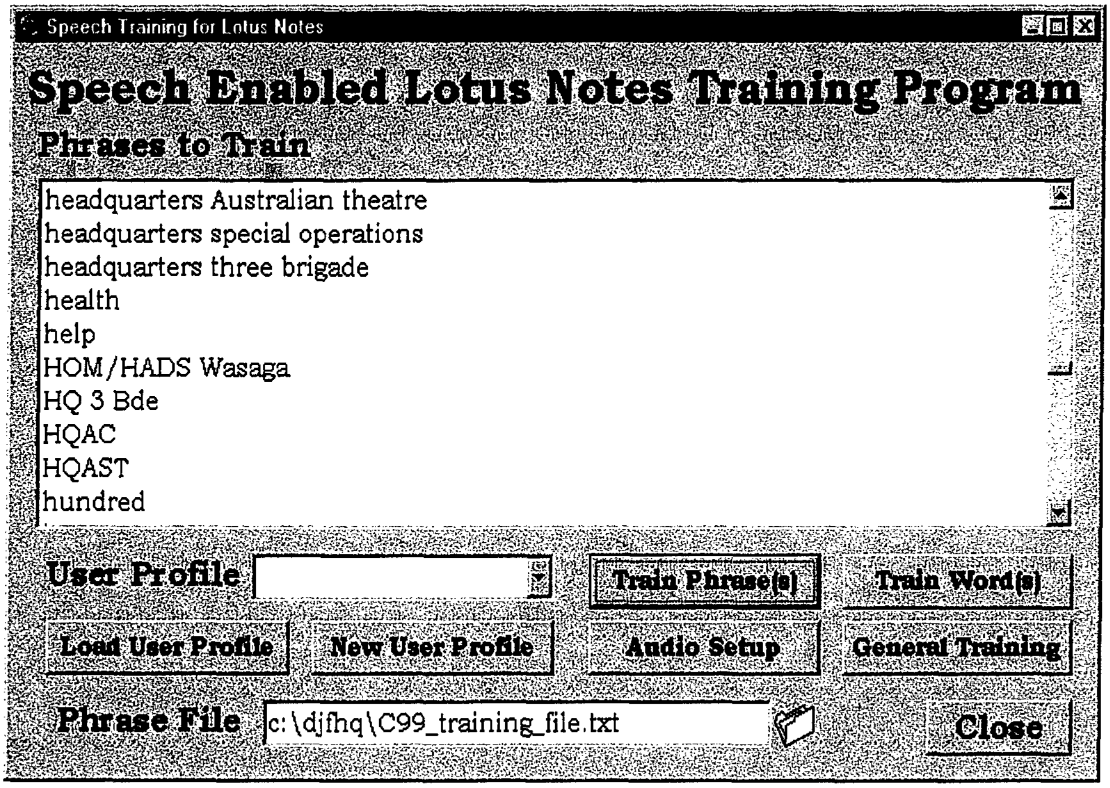
3.3 User Training for a Speech Enabled Lotus Notes Database

In order to increase the recognition accuracy most commercial speech recognisers use application specific syntax (Peckham, 1984). Hence, in order to facilitate user training for data entry using direct speech input and for query from the CSS database, a program was developed in Delphi 4.0. This program uses the Dragon NaturallySpeaking ActiveX components, and allows the user to create new speech profiles and train a custom vocabulary consisting of military acronyms and abbreviations. The document ADFP 123 - Acronyms and Abbreviations has been used to develop a military vocabulary for Dragon NaturallySpeaking. Moreover, each user is able to change how words are pronounced in their vocabulary. Different sub-vocabularies can be used depending on the user's task.

The user training stage for Dragon NaturallySpeaking consists of an Audio Setup and General Training. The Audio Setup calibrates the microphone and adjusts the speech level for optimal performance. During General Training the user reads text from a document so that the recogniser can build phoneme models for the user's voice. The training text can also be changed to suit the user vocabulary requirements. It could, for example, include military acronyms. The time taken for the General Training in Dragon NaturallySpeaking Version 3.52 is 45 minutes. The current Dragon NaturallySpeaking Version 4 takes 7 minutes to train a new user.

The specialised words and acronyms specific to the DJFHQ are stored in a text file, as shown in the 'Phrase File' field in figure 5. Sub-vocabularies can be changed depending on the activity or exercise. The user training with this custom software is identical to the user training in Dragon NaturallySpeaking, but with the additional capability of being able to train specialised words and phrases and military acronyms and abbreviations as provided by the ADF's acronyms dictionary, ADFP 123. The items in the list 'Phrases to Train' can be trained as whole phrases or as separate individual words.

Not withstanding that context insensitivity was a potential problem, it was decided to build prototype models using scripted commands to elicit user requirements. This was as a precursor to the use of ActiveX components as a robust and context sensitive technique to be undertaken at a later stage upon obtaining the requirements. ActiveX controls create and respond to custom events within an application, but they require the source code of the application to be speech enabled.



headquarters Australian theatre
headquarters special operations
headquarters three brigade
health
help
HOM/HADS Wasaga
HQ 3 Bde
HQAC
HQAST
hundred

Phrase File c:\djfhq\C99_training_file.txt

Figure 5: User Training Program for Speech Input to Lotus Notes Command Support System Database

Dragon NaturallySpeaking ActiveX components can be integrated into any Integrated Development Environment (IDE) that supports ActiveX components. These ActiveX components have been tested with the Delphi 4, Visual Basic 5 and Visual J++ 6 IDE's.

4. Integration of English Wizard with Lotus Notes Databases for Data Query

As stated earlier, the second aim of this research project was to improve the user's ability to query the Lotus Notes databases. Operators in DJFHQ wish to query Lotus Notes databases to find documents containing particular keywords, find a staff who has written a document or has been requested for action. The operators may also be interested in obtaining information on staff during shift handovers, or finding statistical information on documents and response times. Hence, in general, three types of improvements were envisioned:

(a) To expand the data search area and resolution of the data, i.e. increase the possible set of queries that can be asked.
(b) To improve human-computer interface for obtaining a user query, i.e. make it easier and/or faster for the operator to ask a question.
(c) To improve the way results from a query are presented to the user. Often when a user queries a database it is an iterative process, so the quicker the user understands the information presented, the quicker the user can enter the next query.

In order to expand the data search area and increase the number of possible queries that can be asked, it was decided to transfer the data into a relational database. A relational database overcomes the limited search capabilities of Lotus Notes. For example, one limitation in Lotus Notes is that the user can only query a database with the results in one of three formats. These formats are:

(a) a single document,
(b) a set of documents, or
(c) highlighted keyword(s) within a document.

Thus it is not possible to have a query such as "How many documents did J6 write?", because the result does not conform to the above formats. A user requiring this information would have to typically ask "What documents did J6 write?" to receive a set of documents and physically count them. Another advantage of having data in a relational database is that it would become possible to compare attributes from different documents and different document types. An example of this is the comparison of when a document was created and when the response to that document was created. This particular type of comparison gives an indication of the response time between an officer receiving an order and that officer acknowledging the status of that order. Such comparisons can be used by the DJFHQ to monitor their internal workflow.

In order to improve the way the operator can enter a query, English Wizard was used to provide a Natural Language Processing capability. In general, the purpose of a Natural Language Processing (NLP) system is to provide a translation service between the human and the computer, where the human computer interaction involves human language expressions (either spoken or typed). In this particular instance the NLP capabilities of English Wizard were used to translate English like expressions to Structured Query Language (SQL) expressions. English Wizard improves human-computer interaction by allowing the user to enter questions as normal English expressions.

In order to improve how the results of a query is presented to the user, an approach was to provide different ways of presenting the data, in particular by using tabular views. Under the current Lotus Notes system data is presented as either a document or as a list of documents. Since a Lotus Notes database can be converted into a relational database, its data can be display in a wide range of views. For example, it is possible to display fields from different document types in the same table; something not possible with the standard Lotus Notes database.

4.1 A Brief Introduction to Natural Language Processing

Generally, natural language processing aims to understand plain spoken English. There are many overviews of Natural Language Processing (NLP), including those by Gasdar and Mellish (1989), Smith (1991), and Bates and Weischedel (1993). Traditional natural language analysis is predominantly syntax-driven where a complete syntactic analysis is performed which attempts to account for all words in an utterance. Such an approach, as reported for example by Bobrow et al. (1990) and Seneff (1992a), while providing some linguistic constraints to the speech recognition component can break down in the presence of unknown words or recognition errors, such as speaker generated artefacts or disfluencies. Besides spoken language tends to be quite informal and ungrammatical. In contrast, others such as Ward (1992) have adopted a semantic-driven approach, deriving a meaning representation by spotting key words and phrases in the utterance. This approach loses the constraint provided by syntax and may not be able to adequately interpret complex linguistic constructs, but provides a more robust strategy that could still answer when a full parse has failed (Jackson et al., 1991; Seneff, 1992b; Stallard and Bobrow, 1992). It appears that for a NLP system, combining syntactic and semantic rules together with (i) a dictionary to help transform the input words into a structure with more meaning, (ii) a domain model of concepts and relationships, and (iii) a task model to provide discourse and pragmatics may provide optimal performance.

4.2 An Overview of Lotus Notes Querying Program

Whilst Lotus Notes does provide a SQL interface via Open DataBase Connection (ODBC) to a database, it is very limited and it was deemed unsuitable. In order to achieve Structured Query Language (SQL) query from a Lotus Notes database, its data has to be moved into a relational database.

The purpose of the Lotus Notes Querying Program was to provide the users of Lotus Notes databases with a higher level of searching capability than what had been available so that they could obtain information and monitor the workflow within their messaging environment. The Lotus Notes Querying Program is composed of several applications integrated together. These applications are:

a) Borland InterBase database, as a relational database, to which Lotus Notes data is copied. InterBase was selected over other database applications such as Microsoft Access because it provided searchable text fields greater than 255 characters. This is important since the Lotus Notes database in use contains a large amount of text in a free text field.
b) English Wizard to provide functional NLP capability by allowing the user to enter a query, as an English expression, either by voice or keyboard. English Wizard generates SQL to access the relational database.
c) Custom software to transfer data from a Lotus Notes database into the InterBase database through use of a text file.
d) Custom software that is incorporated in English Wizard to provide a user interface to enter a query and display tabular results or an aggregate result. A series of questions can be loaded and executed and any question can be saved.
e) Custom software to display the query results within Lotus Notes, since in some cases it was deemed appropriate to display the results in the Lotus Notes format. This software uses the Lotus Notes C++ Application Programming Interface (API) for communicating with Lotus Notes databases. Lotus Notes C++ API provides access to various document variables within a Lotus Notes database. It was used to implement an Agent that would create a Lotus Notes view and insert specified documents that are the result of a query in that view. A Lotus Notes database view is a list of documents that meet some criteria. To create this view the Lotus Notes Querying Program creates a text file of document ID's. These ID's are read by the Lotus Notes Agent, which is activated by the software using the Lotus Notes C++API.

Software routines to extract the data from the Lotus Notes database into the text file were written in LotusScript. The data from the text file is transferred to the InterBase database using software written in Delphi. The section of the Lotus Notes Querying Program for answering questions was written in Delphi too. The program answers questions after establishing an ODBC between English Wizard and the InterBase database as shown later in figure 8.

4.3 Data Extraction from a Lotus Notes Database into an InterBase Database

One of the main reasons for extracting data from Lotus Notes databases as opposed to directly querying those databases was to provide a Structured Query Language (SQL) interface for English Wizard. The data extracting process was done using a combination of several Lotus Notes Agents, written in Lotus Script and custom software. A Lotus Notes Agent can extract data from a single document when the user saves the document. Another manually executed Agent can extract data from all the documents in a selected Lotus Notes database. The result of the extraction is a text file placed in a particular directory. The custom software scans this directory at regular time intervals for text files and inserts the data into an InterBase database. The operating procedures of the DJFHQ states that documents are never deleted, thus it was not necessary to track deletions. The text file contains delimited records that define a value of an object or an attribute of a document in the Lotus Notes database. The fields in the text file are comma delimited, and there are five fields per record. These fields are:

a) the document ID as a unique document identifier to identify the source of the data within the Lotus Notes database
b) the type of object or attribute type of the document
c) the name or identifier of the object or attribute
d) the value of the object or attribute
e) the end of record delimiter. It was necessary to create a record delimiter because the standard record delimiter can appear as part of the Lotus Notes value.

In the Lotus Notes database used in this study, the Log Entry form is a Parent document that is used to request orders to be performed by officers. These officers use the Action document to report the status of the tasks they have been ordered to undertake. The following is an example of a record in the text file describing a Lotus Notes document attribute, identifying the Parent document of an Action document.

```
"AAAFB3E0119325874A25668900133BBA","Document","Parent","2B8091CA011E5328
4A2566890012D3EF","ÈÈÈ"
```

- *"AAAFB3E0119325874A25668900133BBA"* represents a unique document identification and is a 32 character hexadecimal string.
- *"Document"* identifies the next field in the record as a document attribute.
- *"Parent"* signifies the record is about the Parent document.
- *"2B8091CA011E53284A2566890012D3EF"* is the document ID of the Parent document, in this case the Log Entry form.
- *"ÈÈÈ"* is the end-of-record delimiter.

This next example identifies the author of a document in the Lotus Notes database:

```
"01FFB3E0119925874A25668900133AFE","Authors"," 1","CN=SO2 MED/OU=1DIV-
ENOGGERA/OU=LANDCOMD/O=ARMY/C=AU","ÈÈÈ"
```

- *"01FFB3E0119925874A25668900133AFE"* represents a unique document ID and is a 32 character hexadecimal string.
- *"Authors"* means the record contains author information of the particular document.

  " 1" indicates that the author in the author field of the record has firstly edited this document. A " 2" would have indicated that the author in the author field of the record has secondly edited this document.
- *"CN=SO2 MED/OU=1DIV-ENOGGERA/OU=LANDCOMD/O=ARMY/C=AU"* is the author of the record. Thus the author of the document is the S02 army medical officer from 1st Division, Land Command, Enoggera Barracks, Brisbane, Australia.

The final example describes another Lotus Notes document object, in this case the value of a field in a document.

```
"01FFB3E0119925874A25668900133AFE","Field","DATE_LOCAL", 12-FEB-1998","ÈÈÈ"
```

- *"01FFB3E0119925874A25668900133AFE"* represents a unique document ID and is a 32 character hexadecimal string.
- *"Field"* signifies that this record contains the value of a field for the above document.
- *"DATE_LOCAL"* is the name of the field.
- *"12-FEB-1998"* is the contents of the field.

So in the above example the record states that the field *"DATE_LOCAL"* in document *"01FFB3E0119925874A25668900133AFE"* has the value *"12-FEB-1998"*.

In the above examples, the object or attribute value is small in length, but it is possible to get values over 30,000 characters in length, which may be in Rich Text Format (RTF). In a Lotus Notes database, this can occur in the free text field. Thus having a unique end-of-record identifier (in this case "ÈÈÈ") reduces the effort in finding the end of an attribute or object value.

Generally, one text file describes one document, which includes all the document attributes, such as authors, fields' name and contents, and date of creation. The custom software reads and processes the text file and places the data into the InterBase database via an ODBC data source using the InterBase 4.x ODBC driver. This software was written in Delphi because Delphi provided easy integration of software components with InterBase database and hence reduces the development time.

The Lotus Notes Querying Program opens the text file and reads each line until the end-of-record delimiter. The Lotus Notes Querying Program reads the string and inserts the data into the InterBase database. The delimiter is placed not only to indicate the end-of-line, but also the end of a record to ensure that the whole record has been read. It is not secure to remove the delimiter as certain field names like "Body" and "Attachments" hold free text or e-mail messages, hence reading up to an end-of-line would not be sufficient in such cases.

4.4 Data Format Issues in Converting a Lotus Notes Database to an InterBase Database

All the data inserted or updated in the InterBase database was converted to upper case, because the InterBase database is case sensitive. A query like, 'what documents were written about "RED CROSS"' will return records, while a query like, 'what documents were written about "red cross"' will return no records. Any queries entered in the 'Lotus Notes Querying Program' were also converted into upper case.

Lotus Notes Documents employ system date time and Zulu date time formats. An example of system date time is '23/09/98 22:30:02'. The Lotus Notes Querying Program reads this value and separates the date from the time. The time value is converted from 24 hour time to seconds and inserted into the Lotus Notes Querying Program InterBase database.

An example of the Zulu time format is '241322K SEP 98' where:

- '24' = day of month
- 'SEP 98' = month and year
- '13' = hour (24 hour time)
- '22' = minutes.

The Lotus Notes Querying Program reads the Zulu date time from a text file and inserts the Zulu date, time and code respectively into the InterBase database tables. The time component is converted into seconds and entered into the InterBase database.

As for all relational databases, the structure of the tables and the relationship between tables define the set of all possible queries that can be asked from the database. So in this case it was necessary to define and create attributes beyond the single document per record structure of the Lotus Notes database. For example, to calculate the average response time, which is the difference in time between dispatching a document and receiving its response, a table was required that contained both the time the document was created and the time when the Parent document was created. The average response time is converted from seconds to 12-hour time for ease of interpretation. Graphical representation of the allowable types of queries is provided later in this section.

4.5 An Overview of English Wizard

Although there have been recent advances in natural language processing (NLP), only spoken language dialogue systems in limited domains can be envisioned in the near term. A good example of this is question answering domains such as database interfaces where the level of performance the system is regarded as being acceptable (Bates, 1994). Our aim was to improve the user's ability to enter a query by allowing the user to enter queries as English expressions. In order to meet this aim we needed to convert English like queries (either spoken or typed) into a format that could be applied to a database.

English Wizard is a NLP software product that translates English like queries into Structured Query Language (SQL). The SQL can be used directly as a query into a relational database. English Wizard cannot translate every query put to it. The SQL English Wizard produces must be executable in the information domain of the relational database. Therefore a user query must result in obtaining either a collection of fields within a database or some quantitative functions (such as a total or average value) on a collection of fields. In summary, the set of queries English Wizard can answer is limited by the structure of the database it is connected to. The structure of the InterBase database that was created for this work from the Lotus Notes database is shown in figure 6.

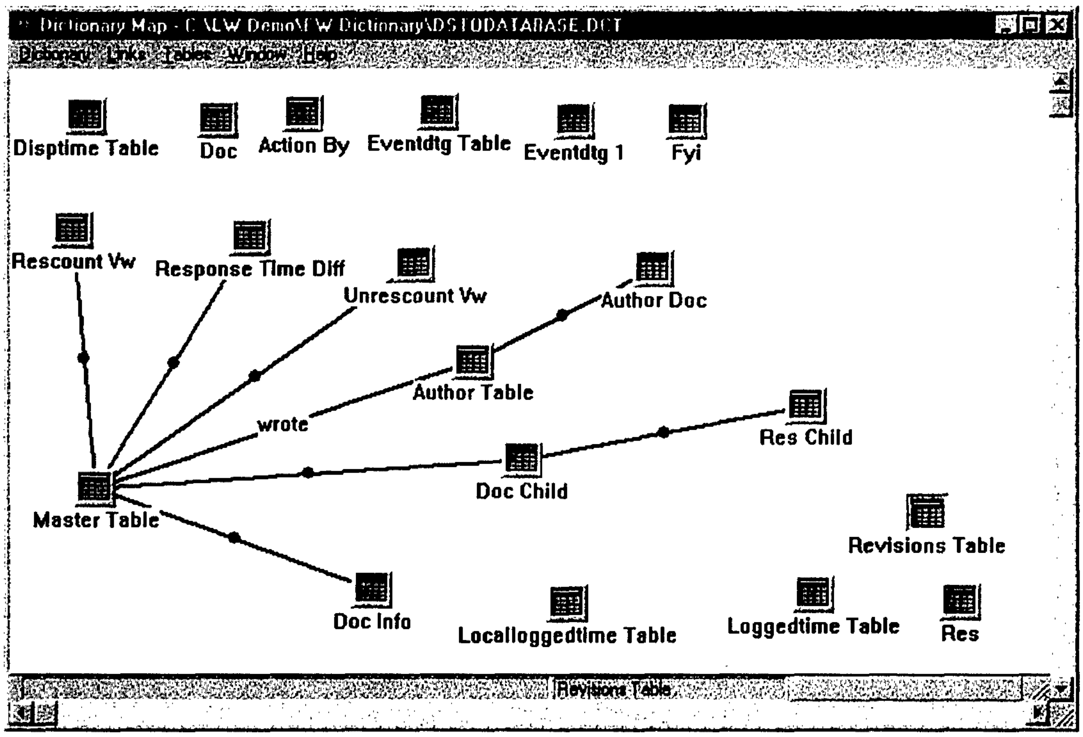


Figure 6: English Wizard Map showing the structure of an InterBase database

4.5.1 English Wizard Dictionary

As with most NLP products, English Wizard uses a combination of syntactic (grammar) and semantic (meaning) rules to perform translations. English Wizard obtains syntactic and semantic information on the database via a document called "Dictionary", which the application developer populates with the appropriate information. The English Wizard Dictionary obtains some syntactic information automatically by using an ODBC connection to the relational database and accesses the database's meta-data, such as table names and structures, column names and fields and, in some cases, joins between tables. The application developer provides some syntactic information such as joins missed by the automatic system, but the developer provides mainly semantic information. This semantic information is usually in the form of "meaningful" key-word labelling various database structures. For example, labelling a table "salesman", labelling another table "customer" and labelling the join between these tables "sell to" defines a rule that "salesman sell to customers". In figure 6 the label 'wrote' between the Master Table and the Author Table defines a rule identifying 'author wrote document' (see English Wizard Dictionary Administrator's Guide and English Wizard Programmer's Reference). The semantic information entered in English Wizard Dictionary could by classified into approximately 30 various elements, some of these are:

a) Labels - These are meaningful names of various database elements.
b) Functions - The definition of calculations, for example age = today_date – date_of_birth, where today_date and date_of_birth are field names.
c) Synonyms - These are alternative names for functions and labels, eg "Product" and "Stock" have the same meaning.
d) Root words - Root words are defined as a field or group of fields in a particular table, or null words such as "a" and "the". For example the root word "Customer" is defined within the dictionary as the Customer.firstname and the Customer.lastname. As another example, the root word 'documents' is defined within the English Wizard dictionary to represent document IDs. The answer to the question 'who wrote documents about "RED CROSS"' includes document IDs. This is so that any reply to queries involving the word document returns the document ID.
e) Groups - A group define a field or group of fields where their table is not defined. For example, the group "who" defines fields "firstName" and "lastName" for any table that has both those fields. So the query "who sold the most product" would return the salesman's first and last name, and the query "who ordered the most product" would return the customer's first and last name.

English Wizard Dictionary cannot provide all the semantic information. Where there are gaps in the translation of a particular query, English Wizard may prompt the user for additional information; this information can be saved in the English Wizard Dictionary.

Figure 7 illustrates the English Wizard Dictionary. The window of English Wizard shown is used to view and edit the links between the tables.

In figure 6, the relationship defined by the word 'wrote' between the tables called Master Table and Author Table associates document identification values. It is possible to invoke the English Wizard map from the Lotus Notes Querying Program by entering a question where English Wizard is required to interpret it. Questions such as 'who wrote documents about "RED CROSS"' obtain its information from within the Master Table and Author Table. The word 'who' is defined in the dictionary, shown in figure 7, as an author in the Author Table.

The SQL created by English Wizard from the question 'who wrote documents about "RED CROSS"' is provided as below:

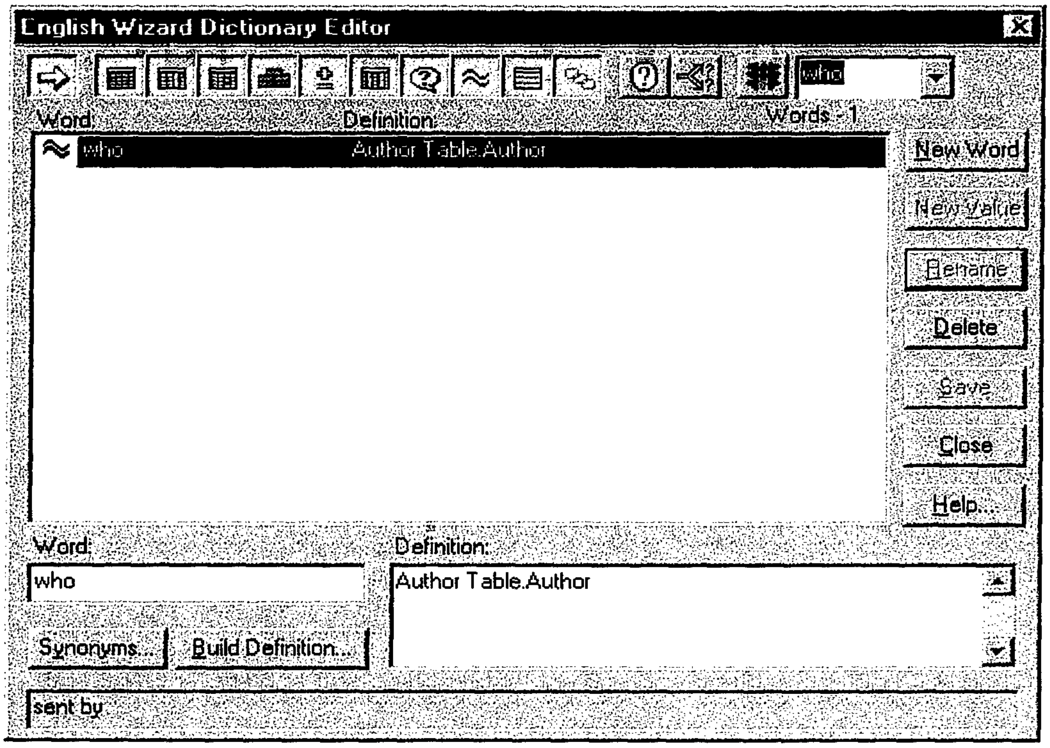
```
SELECT DISTINCT AUTHOR_TABLE.AUTHOR, MASTER_TABLE.DOC_ID,
SUBJECT
FROM  AUTHOR_TABLE , MASTER_TABLE
WHERE (MASTER_TABLE.DOC_ID is not NULL  and  SUBJECT  like '%RED
CROSS%') and AUTHOR_TABLE.DOC_ID=MASTER_TABLE.DOC_ID
```

'Who wrote about RED CROSS' is a query illustrating how English Wizard may prompt the user for additional information, thus modifying the SQL of the query that English Wizard produces. The phrase 'wrote about' is defined to represent the fields in the Master Table called 'Body' and 'Attachments'. English Wizard would prompt the user on whether "RED CROSS" is located in either of these fields. Below is the English Wizard SQL from answering 'yes' to whether "RED CROSS" was a 'wrote about' and that the selection was in the 'Attachments' and 'Body' fields.

```
SELECT DISTINCT AUTHOR_TABLE.AUTHOR,  ATTACHMENTS ,  BODY
FROM  AUTHOR_TABLE ,  MASTER_TABLE
WHERE (( ATTACHMENTS  like '%RED CROSS%' and  BODY  like '%RED
CROSS%')) and AUTHOR_TABLE.DOC_ID=MASTER_TABLE.DOC_ID
```

4.6 Connectivity between English Wizard, InterBase Database and Lotus Notes Querying Program

The conversion from an English query to the resulting data is a two stage process. During the first stage, English Wizard receives an English like query via an ODBC connection to the Lotus Notes Querying Program. In the second stage English Wizard converts this expression to SQL and extracts the result data from the InterBase database. This is done via an ODBC connection between the InterBase database and English Wizard. English Wizard then passes the results of the query through an ODBC connection to the Lotus Notes Querying Program.



English Wizard Dictionary Editor

Word: who

Definition: Author Table.Author

sent by

Figure 7: English Wizard Dictionary

These stages are shown in figure 8 which illustrates components of the Lotus Notes Querying Program. It is possible to request the SQL generated by English Wizard. This is mainly used by the application developer to verify the translation from an English query to SQL and is seldom shown to the user.

4.7 List of Software Components

Commercial Software used by the system or used during the development of the system are:

Borland Delphi for Windows 95 and Windows NT version 4 by Inprise Corporation
English Wizard version 3 by Linguistic Technology Corporation and its Driver 32 as the ODBC Driver
Lotus Notes C++ API Release 4.12 (see Lotus Notes C++ API 4.12 at www.lotus.com)
Lotus Notes Client version 4.6 by Lotus
InterBase version 4.0 by Inprise Corporation, InterBase ODBC Driver, and InterBase Windows ISQL.

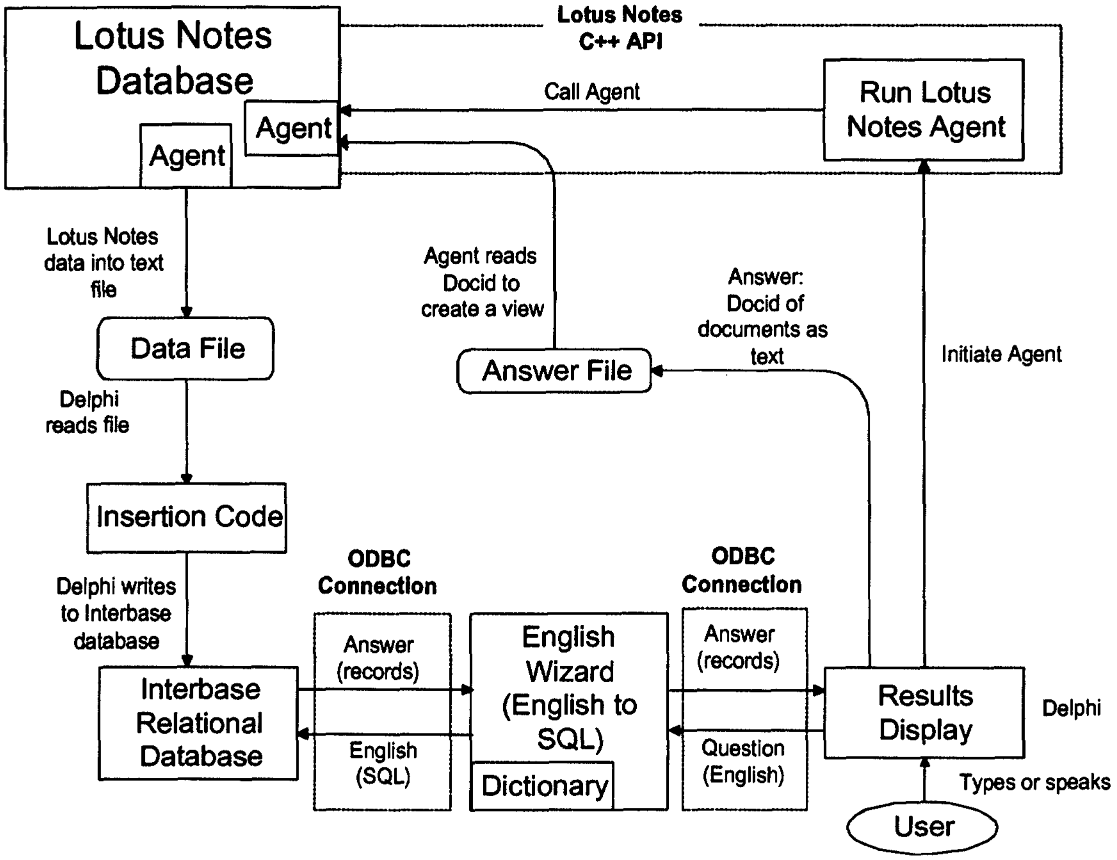
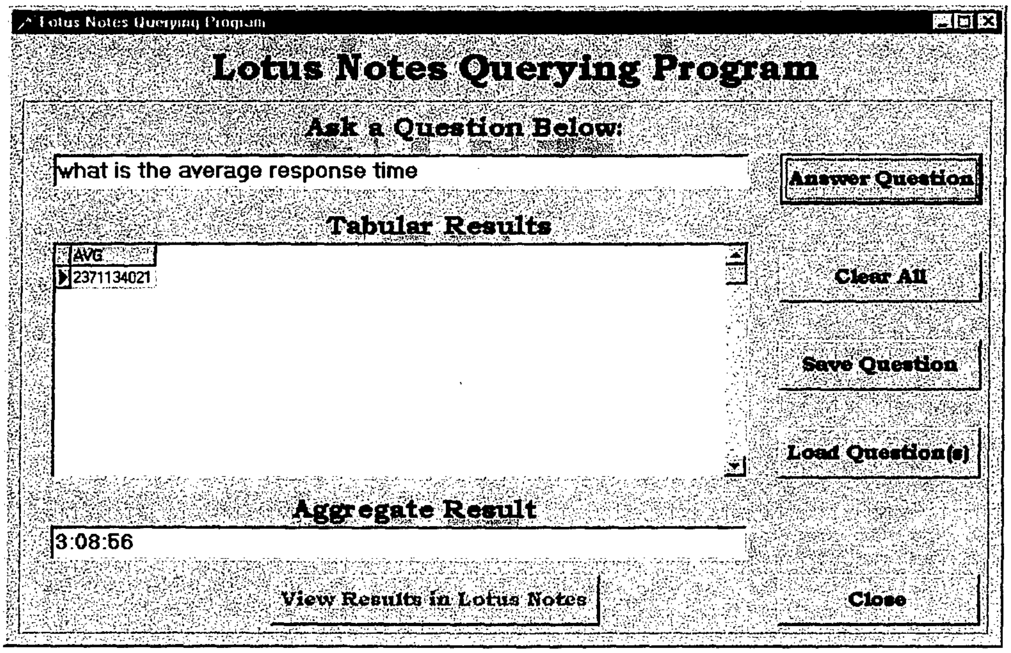


Figure 8: A Block Diagram Showing Components of Lotus Notes Query Program

4.8 Querying Process

The answer to a question is displayed in the main interface of the Lotus Notes Querying Program and can be as a value or in the form of document(s). Questions like "What is the average response time?" will return a value, whereas questions like "who wrote documents about "RED CROSS" will return records in the form of a list of respective authors, document identification values and subjects (see figures 9a and 9b).

The answer can also be represented in the form of a Lotus Notes database view. The user can open, read and modify the data in any document within the Lotus Notes view. On closing a modified document, the agent will create a text file of that document. The Lotus Notes Querying Program will update the document data in the Lotus Notes Querying Program InterBase database. The document will also be updated in the Lotus Notes database.



Lotus Notes Querying Program

Ask a Question Below:

what is the average response time

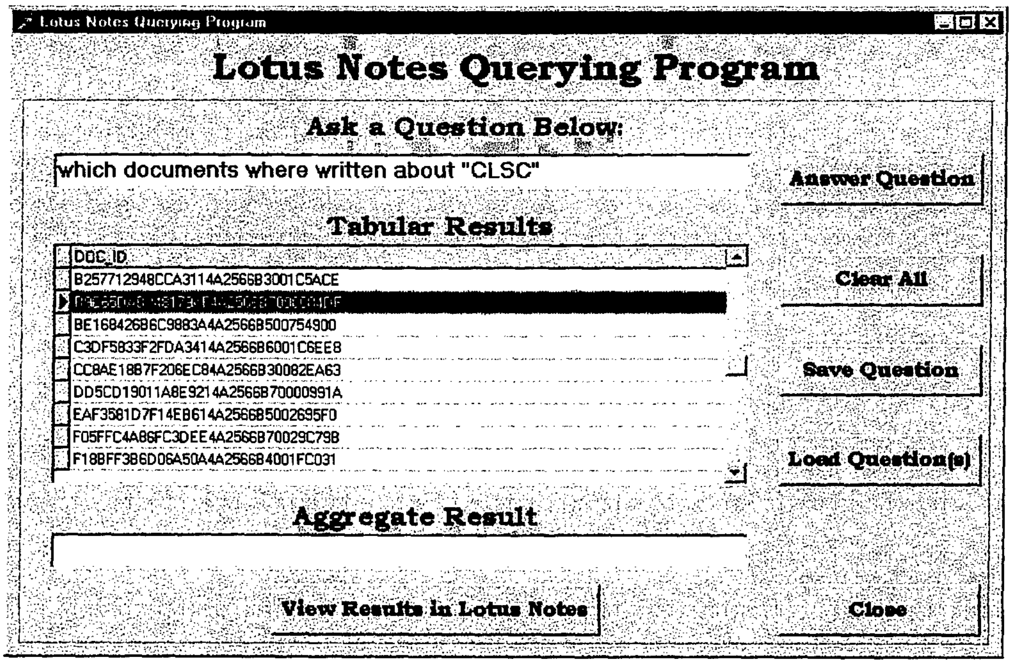
Tabular Results

AVG
2371134021

Aggregate Result

3:08:56

(a)



Lotus Notes Querying Program

Ask a Question Below:

which documents where written about "CLSC"

Tabular Results

DOC_ID
B257712948CCA3114A2566B3001C5ACE
[illegible]
BE168426B6C9883A4A2566B500754900
C3DF5833F2FDA3414A2566B6001C6EE8
CC8AE18B7F206EC84A2566B30082EA63
DD5CD19011A8E9214A2566B70000991A
EAF3581D7F14EB614A2566B5002695F0
F05FFC4A86FC3DEE4A2566B70029C79B
F18BFF3B6D06A50A4A2566B4001FC031

Aggregate Result

(b)

Figure 9: Results of Two Allowable Lotus Notes Queries. As Shown, Result from a Lotus Notes Query is Returned as a Value (a) or as Records (b).

5. Discussion

The integration of both speech recognition and database query technologies as described has raised many issues. Some of these issues are technical and will be addressed with continued development of the technological concepts. However, other issues are of a socio-technological nature. These have arisen from the military domain for which these concepts have been prototyped and developed. In order to produce an effective user interface with a speech recognition capability, studying human to human conversations in the application domain is required (Yankelovich et al., 1995). The issues of concern include developing a vocabulary of military acronyms and jargons, user interface design for communication by speech, and system scalability. Addressing all these issues will be paramount in the development of speech recognition and database querying technologies for future military messaging environments.

Lea (1982) produced an extensive list of over 80 factors, which might influence the performance of a speech recogniser. Language factors are important in integrating speech recognition with Lotus Notes messaging systems at the DJFHQ. The Australian Defence Force (ADF) produces and maintains a document, called ADFP 123 - Acronyms and Abbreviations, which contains the definitions of all the acronyms and abbreviations used within the ADF. As an entry in this document the acronym 1 RAR is defined to represent the 1st Royal Australian Regiment. Military personnel from the DJFHQ pronounce this acronym as "1 r a r" which, of course, is quicker than saying it in full. The staff at the DJFHQ also omit pronouncing the slash in the company names. For example, the acronym 10/27 RSAR stands for the 10th/27th Royal South Australian Regiment. This acronym is spoken as '10 27 r s a r'. The staff pronounce the acronym how it is spelt. For example the acronym CONOPS, representing Concept of Operations, is spoken as 'con ops' and not 'c o n o p s'. An important aspect of this is to provide information on how the users pronounce specialised words or utter acronyms and jargon to others and to the speech user interface. Knowledge of how ADF staff pronounce acronyms will remain crucial in developing vocabularies to deliver a speech recognition capability to military messaging.

The unpredictability of the technology continues to result in user frustration (Ellis, 1999). A spoken utterance can be correctly recognised once, but misrecognised when spoken a second time due to variabilities in speech. Therefore it is difficult for the user to maintain a conceptual model of the application's behaviour. However, accurate domain specific vocabularies will result in less user frustration by limiting user vocabulary development and error correction (Noyes and Frankish, 1989).

Speaking rate is another factor that can adversely affect the performance of a speech recogniser. Speaking rate is a function of the size of the active vocabulary in the RAM and the CPU speed. The larger the vocabulary size, the larger the system response time will become (Kloosterman 1994). A typical commercially available product on the market can support up to 60,000 words active in RAM and contains a back-up dictionary of 270,000 words. Military applications often place higher demand on robustness to ambient and environmental noise and user stress than do civilian applications, but military applications can often be carried out in constrained task domains, where, for example, the active vocabulary and grammar for speech recognition can be limited (Weinstein, 1994).

It is interesting to note that the rate and manner of speaking and the degree of co-articulation varies between human-to-human and human-to-computer interaction. Users of continuous speech recognisers adjust their speaking rate to improve the recognition performance. Users normally slow down their speaking rate and articulate words more carefully. It is also noted that disfluencies or speaker-generated artefacts, such as tongue clicks, 'ums', 'urs' and coughs are less evident when the user is interacting with a computer than with another human.

The success of a speech recognition capability within a military messaging environment will also depend on the success of research into training and enrolment, error detection and correction techniques (Noyes and Frankish, 1989). The enrolment text used for the Dragon NaturallySpeaking software was a selection that Dragon Systems provides from Arthur C. Clarke "3001 – The Final Odyssey". It is possible to change this training text to be more relevant to the application domain, thus increasing the recognition rate for the users. As such, knowledge of the application domain will in many ways influence the success of a speech recognition capability in military messaging environments.

The ease of error correction and recovery determines the user acceptance of a speech recogniser. In a recent study (Karat et al., 1999) the most common command used by users of continuous speech recognisers was the command that reversed the immediately preceding action. In this study the average correction time for words and phrases was three times as long as dictating those words or phrases. Therefore, although it would often be most efficient to enter data by a combination and speech and manual input modalities, nevertheless error correction should be done through either manual input alone or through an interactive correction mechanism by switching between modalities (Suhm et al., 1999).

When the Lotus Notes Query Program creates a view in Lotus Notes as a result of user querying the system, the view is not updated in the Lotus Notes Client. The user is forced to close and reopen the Lotus Notes Client in order to re-display the view created as a result of a query. This problem will be solved by using Java to communicate with the Lotus Notes databases via Lotus Notes Java classes.

While the Lotus Notes Query Program was built as a concept demonstrator, it is worth noting the scalability of the technology used in the system. The system as described exports the Lotus Notes data to a text file that is imported into an InterBase relational database. This approach would not be feasible for a large scale distributed system due the slow speed of the Lotus Notes agent in writing the text file. In addition, importing data from the text file to a relational database may involve considerable processing time.

A second problem with the text file approach is one of security. Both Lotus Notes and (most) relational databases have control mechanisms to prevent unauthorised access to the data they contain. A text file has no control mechanisms and is easily read. The use of Java, namely its Java DataBase Connection (JDBC) to communicate directly with a Lotus Notes database via the Java classes, will also solve this problem.

Another scalability issue relates to the relational database itself. Since the data in the relational database mirrors the data in the Lotus Notes, the size and complexity issues in the Lotus Notes data may be passed on to the relational database. The relational database is designed for the types of queries the users require. Thus, the more complex the Lotus Notes database is, the more difficult designing the relational database will become.

During a visit to the DJFHQ, military users trained the Dragon speech recogniser and performed some data entry by speech. The vocabulary they used contained the standard ADF acronyms and abbreviations. Speech materials recorded from the military users has formed a corpus of speech data. These will later be used to conduct experiments for system performance assessment.

Users first impression about interacting with a computer using speech in terms of speed and accuracy were very positive. However, one of the most important aspects of this research work is a comprehensive user evaluation of the system through a human computer interaction study that is yet to be carried out. The study will require both qualitative and quantitative performance analyses. Preliminary observations indicate that speech integration can potentially provide a more natural, faster and more accurate means of user data entry in a command and control environment. The new query interface allows the user to ask a broader range of questions from a Lotus Notes database than before. This has shown promise to improve the efficiency of users in obtaining information or monitoring an operation or a request in progress. It will however be very hard to perform quantitative evaluation of user interaction with a spoken language query or dialogue system due to inherent variabilities present in asking English questions.

6. Future Directions

Future research and development of the concept demonstrator described in this report will be done using the Java programming language to improve the integration of speech technology with Lotus Notes databases. Improving the user interface and the mechanisms for querying Lotus Notes databases will also be attempted. In order to create new entries in a Lotus Notes database, the Lotus Notes database Client is used.

It is envisaged that a user interface for a Lotus Notes database that allows the user to create, edit and view entries in the database will be produced. The user will be able to enter information by voice as well as keyboard and mouse. An improved searching capability will also be provided to the user. Internet browsers (e.g. Internet Explorer 4.0) are being used at the DJFHQ to read Lotus Notes databases. Since this GUI will be written in Java it can be used within an Internet browser such as the Internet Explorer 4.0.

Naturally, a user-orientated speech recognition interface improves the useability of that interface (Kloosterman, 1994). For natural and efficient speech user interfaces careful attention must be applied to the transaction and dialogue design (Peckham, 1984). The design of a speech interface must consider the interrelationship among the task requirements, technological capabilities and user expectations in implementing an interface that will facilitate "ease of use" and naturalness of human-machine interaction. A speech interface can be improved by using directive prompts and confirmation protocols. These prompts could be visual or auditory by incorporating text to speech synthesis. (Peckham, 1984). Furthermore, providing user control to access instructions and correct errors would provide further enhancement (Kamm, 1994; Yankelovich et al., 1995). The use of Java will allow the speech recognition software to provide visual and auditory feedback mechanisms to aid error recovery and correction.

Lotus Notes does not allow for an Open DataBase Connection (ODBC) to a database. Therefore in order to achieve Structured Query Language (SQL) query from a Lotus Notes database, one approach could be to use the Lotus Notes C++ API to move the data into a relational database. Data communication with Lotus Notes can be achieved by using the Lotus Notes C++ API; such an API is also available for Java. It is intended to use Java API to implement the speech integration with the Lotus Notes graphical user interface as an improvement to the current prototype. This research and development will explore the use of Java to design a speech user interface and to improve the user capabilities in querying and searching Lotus Notes databases through the user interface.

7. Conclusions

The language and database technologies described in this report combine to form a concept demonstrator of how various filled in forms from a Lotus Notes database can be filled through direct speech input and how natural language queries can be made from a Lotus Notes database.

Dragon NaturallySpeaking version 3.52 has been integrated with a Lotus Notes database known as the Command Support System database. Log operators from the Logistics and Administration Cell (J14) within the Deployable joint Force Headquarters have been trained to use the speech recogniser for data entry. In doing so information about how acronyms are spoken has been elicited and clues have been gathered as to when, why and how the operators use the Lotus Notes databases. Speech integration has provided flexibility and naturalness in user interaction with the forms within the CSS database and has indicated that direct speech input can provide a potentially faster and more accurate means of user interaction with messaging and communication systems.

The use of ActiveX component for the integration of speech recognition with Lotus Notes will be further explored. However, the main focus will be on exploring the use of Java to create a custom speech enabled user interface to a Lotus Notes database so that speech integration becomes robust and context sensitive. Context awareness results in a higher recognition accuracy and better error recovery.

The query interface to the Lotus Notes database has enabled a user to ask questions in plain English and obtain documents containing certain information or numerical values related to a certain action and response. Although the possible range of queries is limited, the prototype model has thus far enabled the operators at the DJFHQ to monitor their internal workflow in a flexible and efficient manner. It has improved resolution of the data that can be searched, has reduced the response time of the system, but has allowed operators to ask a broader range of questions than they had been able to ask from the Lotus Notes databases.

There is a delay for the concept demonstrator as described in this report to update a document view in Lotus Notes Client since the user has to close and reopen the Lotus Notes Client to re-display the view created as a result of a query. Additionally, limited scalability of the commercial software technology used and the security implications of using a text file need to be addressed in further developing the system. The language technology concept demonstrator will attempt to improve the querying and data extraction from a Lotus Notes database to a relational database in order to meet the client technology and security requirements. It is intended that a natural language query capability with a Lotus Notes database via a Java based user interface will be integrated in the concept demonstrator so as to improve user querying and searching the databases.

8. References

Bates, M. (1994) "Models of natural language understanding", In Voice communication between humans and machines, D.B. Roe and J.G. Wilpon (Eds.), National Academy of Sciences, pp. 238-253.

Bates, M. and Weischedel, R.M. (1993) (Eds.) "Challenges in natural language processing", Cambridge University Press, Cambridge.

Borrow, R., Ingria, R. and Stallard, R. (1990) "Syntactic and semantic knowledge in the DELPHI unification grammar", Proceedings of DARPA speech and natural language workshop, pp. 263-265.

Cupples, E.J. and Beek, B. (1990) "Applications of audio/speech recognition for military applications", In Proceedings of the NATO/AGARD lecture series No. 170, speech analysis and synthesis and man-machine speech communications for air operations, pp. 8-1-8-10.

Dragon Systems (1997) "DragonDicate Macro Language - Guide and Reference", Dragon Systems Inc., Second Edition.

Dragon Systems (1998) "Dragon NaturallySpeaking ActiveX Components – Guide and Reference", Dragon Systems Inc., Version 1.0.

Ellis, D. (1999) "Dragon NaturallySpeaking Preferred", Australian Personal Computers, March 1999, p. 42.

"English Wizard (1997) "Dictionary Administrator's Guide", Linguistic Technology Corp., Littleton, MA, USA.

"English Wizard (1997) "Programmer's Reference", Linguistic Technology Corp., Littleton, MA, USA.

Flanagan, J.L. et al. (1984) "Automatic speech recognition in severe environments", National Research Council, Committee on computerized speech recognition technologies, Washington, D.C., National Academy Press.

Gasdar, G. and Mellish C. (1989) Natural language processing in LISP, Addison-Wesley, Reading, Mass.

Jackson, E., Appelt, D., Bear, J., Moore, R. and Podlozny, A. (1991) "A template matcher for robust NL interpretation", Proceedings of DARPA Speech and Natural Language Workshop, pp. 190-194.

Kamm, C. (1994) "User interfaces for voice applications", In Voice communication between humans and machines, D.B. Roe and J.G. Wilpon (Eds.), National Academy of Sciences, pp. 422-442.

Karat, C-M, Halverson, C., Horn, D. and Karat, J. (1999) "Patterns of entry and correction in large vocabulary continuous speech recognition systems", Proceedings of Computer Human Interaction '99, pp. 568-575.

Kloosterman, S.H. (1994) "Design and implementation of a user-orientated speech recognition interface: the synergy of technology and human factors", Interacting with Computers, Vol. 6, No. 1, pp. 41-60.

Lea W.A. (1982) "What causes speech recognisers to make mistakes?", IEEE ICASSP, Paris, France, pp. 2030-2033.

Lea W.A. (1983) Selecting the best speech recogniser for the job", Speech Technology, Vol. 1, No. 4, pp. 10-29.

Lotus Notes (1999) "Lotus Notes C++ API 4.12", The Lotus Notes Developer Central, available at http://www.lotus.com,
Help is available at http://www.lotus.com/rw/devchat.nsf/by+Category?OpenView

Makhoul, J. Crystal, T.H., Green, D.M., Hogan, D., McAulay, R.J., Pisoni, D.B., Sorkin, R.D. and Stockham, Jr., T.G. (1989) "Removal of noise form noise-degraded speech signals", National Research Council, Committee on Hearing, Bioacoustics, and Biomechanics, Washington, D.C., National Academy Press.

Martin T.B. and Welch J.R. (1980) "Practical speech recognisers and some performance effectiveness parameters (part 1 – motivation and general review)". In Trends in speech recognition, W.A. Lea (Ed), Prentice-Hall, NJ, USA, pp. 24-38.

Noyes, J.M. and Frankish, C.R. (1989) "A review of speech recognition applications in the office", Behaviour and Information Technology, Vol. 8, No. 6, pp. 475-486.

Peckham, J.B. (1984) "Automatic speech recognition - a solution in search of a problem?", Behaviour and Information Technology, Vol. 3, No. 2, pp. 145-152.

Seneff, S. (1992a) "TINA: A natural language system for spoken language applications", Computational Linguistics, Vol. 18, No.1, pp. 61-86.

Seneff, S. (1992b) "Robust parsing for spoken language systems", Proceedings ICASSP, pp.189-192

Smith, G.W. (1991) "Computers and human language", Oxford university Press, Oxford, U.K.

Stallard, D. and Bobrow, R. (1992) "Fragment processing in the DELPHI system", Proceedings of DARPA Speech and Natural Language Workshop, pp. 305-310.

Suhm, B., Myers, B., and Waibel, A. (1999) "Model-based and empirical evaluation of multimodal interactive error correction", Proceedings of Computer Human Interaction '99, pp. 584-591.

Ward, W. (1990) "The CMU air travel information service: Understanding spontaneous speech", Proceedings of DARPA Speech and Natural Language Workshop, pp. 127-129.

Weinstein, C.J. (1994) "Military and government applications of human-machine communication by voice". In Voice communication between humans and machines, D.B. Roe and J.G. Wilpon (Eds.), National Academy of Sciences, pp. 357-370.

Woodard, J.P. and Cupples, E.J. (1983) "Selected military applications of automatic speech recognition, IEEE Communication Magazine, Vol. 21, No. 9, pp. 35-44.

Yankelovich, N., Gina-Anna, L. and Marx, M. (1995) "Designing SpeechActs: Issues in speech user interfaces", Proceedings of Computer Human Interaction '95, ACM.

Zue, V. (1994) "Human computer interaction using language based technology", Proceedings of IEEE International Symposium on Speech, Image Processing and Neural Networks, pp. I-VII.

Application of Spoken and Natural Language Technology to Lotus Notes Messaging and Communication

Stepahne Lajoie, Oliver Carr, Michael Coleman and Ahmad Hashemi-Sakhtsari

(DSTO-TR-1265)
DISTRIBUTION LIST

AUSTRALIA

		Number of Copies
DEFENCE ORGANISATION		
Task sponsor:		
Comd DJFHQ		1
HKS		1
S&T Program		
Chief Defence Scientist	)	1 shared copy
FAS Science Policy	)	
AS Science Corporate Management	)	
Director General Science Policy Development		1
Counsellor, Defence Science, London		Doc Control Sheet
Counsellor, Defence Science, Washington		Doc Control Sheet
Scientific Adviser to MRDC Thailand		Doc Control Sheet
Scientific Adviser - Joint		1
Navy Scientific Adviser		1 copy of Doc Control Sheet 1 copy and 1 distribution list
Scientific Adviser - Army		Doc Control Sheet and 1 distribution list
Air Force Scientific Adviser		1
Director Trials		1
Aeronautical & Maritime Research Laboratory		
Director		1
Electronics and Surveillance Research Laboratory		
Chief Information Technology Division		1
Research Leader Command & Control and Intelligence Systems		1
Research Leader Military Information Enterprise		1
Research Leader Human Factors, Land Operations Division		1
Head, Systems Simulation and Assessment Group		Doc Control Sheet
Head, Distributed Systems Group		Doc Control Sheet
Head, C3I Systems Concepts Group		1

Head, Enterprise Visualisation Instrumentation and Synchronisation Group	Doc Control Sheet
Head, Intelligence Group	Doc Control Sheet
Head, C3I Operational Analysis Group	Doc Control Sheet
Head Information Exploitation Group	1
Head, Human Systems Integration Group	1
Head, C2 Australian Theatre	1
Task Manager, Ahmad Hashemi-Sakhtsari, (HSI Group)	1
Authors	4
Publications and Publicity Officer ITD/Executive Officer ITD	1

DSTO Library and Archives

Library Fishermens Bend	Doc Control Sheet
Library Maribyrnong	Doc Control Sheet
Library Edinburgh	2
Australian Archives	1
Library, MOD, Pyrmont	Doc Control Sheet
US Defense Technical Information Center,	2
UK Defence Research Information Centre,	2
Canada Defence Scientific Information Service,	1
NZ Defence Information Centre,	1
National Library of Australia,	1

Capability Systems Staff

Director General Maritime Development	Doc Control Sheet
Director General Land Development	Doc Control Sheet
Director General Aerospace Development	Doc Control Sheet

Knowledge Staff

Director General Command, Control, Communications and Computers (DGC4)	Doc Control Sheet

Navy

SO (Science), COMAUSNAVSURFGRP, Bld 95, Garden Island, Locked Bag 12, Pyrmont NSW 2009.	Doc Control Sheet

Army

Chief of Staff, DJFHQ, Enoggera, Qld 4051	1
DC2I, Knowlegde Systems	1
Lieutenant Colonel Norm Cognet, Staff Officer Aviation, DJFHQ, Enoggera, Qld 4051	1
SO (Science), DJFHQ, (L) MILPO, Enoggera, Qld 4051	1
Stuart Schnaars, ABCA Standardisation Officer, Tobruk Barracks, Puckapunyal 3662	4

Intelligence Program

DGSTA Defence Intelligence Organisation	1
Manager Information Centre, Defence Intelligence Organisation	1

Corporate Support Program

Library Manager, DLS-Canberra	1

Universities and Colleges

Australian Defence Force Academy	1
Library	1
Head of Aerospace and Mechanical Engineering	1
Serials Section (M list)), Deakin University Library	1
Hargrave Library, Monash University	Doc Control Sheet
Librarian, Flinders University	1
Professor Robin King, Pro Vice Chancellor, Division of Information Technology, Engineering and the Environment, University of South Australia	1

Other Organisations

NASA (Canberra)	1
AusInfo	1
State Library of South Australia	1

OUTSIDE AUSTRALIA

Abstracting and Information Organisations

Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts US	1
Documents Librarian, The Center for Research Libraries, US	1

Information Exchange Agreement Partners

Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1

SPARES	5
Total number of copies:	**63**

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION **DOCUMENT CONTROL DATA**	1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE Application of Spoken and Natural Language Technologies to Lotus Notes Based Messaging and Communication	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)
4. AUTHOR(S) Stephane Lajoie, Oliver Carr, Michael Coleman and Ahmad Hashemi-Sakhtsari	5. CORPORATE AUTHOR Electronics and Surveillance Research Laboratory PO Box 1500 Edinburgh SA 5111 Australia

6a. DSTO NUMBER DSTO-TR-1265	6b. AR NUMBER AR-012-130	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE January 2002

8. FILE NUMBER N9505/23/29	9. TASK NUMBER 98/288	10. TASK SPONSOR COMD DJFHQ	11. NO. OF PAGES 45	12. NO. OF REFERENCES 32

13. URL ON THE WORLD WIDE http://www.dsto.defence.gov.au/corporate/reports/DSTO-TR-1265.pdf	14. RELEASE AUTHORITY Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

Voice data processing
Message processing
Groupware
Human computer interface
Speech recognition

19. ABSTRACT

Speech can potentially provide a natural, efficient and flexible means of user interaction with computers in a messaging and communication environment. This report presents a spoken language data entry and database query concept demonstrator. Firstly, the report discusses a technique for integrating a commercial large-vocabulary continuous speech recogniser with a Lotus Notes database user interface for data entry into Lotus Notes forms. Secondly, the report describes the integration of a number of software components, which include a commercial natural language processing product, in order to produce a query interface to access a Lotus Notes database. The military operators can query a database in plain English to obtain information and to monitor their internal workflow.

TECHNICAL REPORT DSTO-TR-1265 AR-012-130 JANUARY 2002

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

ELECTRONICS AND SURVEILLANCE RESEARCH LABORATORY
PO BOX 1500 EDINBURGH SOUTH AUSTRALIA, 5111
AUSTRALIA, TELEPHONE (08) 8259 5555